УДК 004.934.1'1

В.Ю. Шелепов, А.В. Ниценко, А.В. Жук
Институт проблем искусственного интеллекта МОН Украины и НАН Украины, г. Донецк
shel@iai.dn.ua, nav_box@mail.ru, juk@iai.dn.ua

Построение системы голосового управления компьютером на примере задачи набора математических формул

Статья посвящена описанию модульной системы голосового управления компьютером. Рассматриваются вопросы, связанные с архитектурой такой системы, структурой управляющего модуля и форматом словаря команд, приводится описание алгоритмов, лежащих в основе подсистемы распознавания речи. Рассматривается подход к обучению системы для конкретного диктора, предполагающий обучение в процессе использования системы.

Введение

Одним из направлений интеллектуализации компьютерной обработки информации является распознавание речи, т.е. разработка методов и алгоритмов интерпретации звуковых сигналов, позволяющих компьютеру определить в анализируемом звуковом сигнале наличие речевых данных и правильно прореагировать на них. Работы по распознаванию речи, помимо всего прочего, ведут к созданию части речевого интерфейса «человек – компьютер», отвечающей за обработку данных от пользователя, т.е. систем голосового управления. **Целью данной работы** является описание модульной системы голосового управления компьютером на основе распознавания раздельно произносимых команд и иллюстрация возможностей такой системы в приложениях для набора математических формул, входящих в состав пакета Microsoft Office.

Архитектура системы голосового управления

Описываемая в данной работе система голосового набора математических формул проектировалась и разрабатывалась как часть системы голосового управления компьютером. Программа имеет модульную структуру: вокруг основного приложения – распознавателя речи, реализующего взаимодействие с пользователем и функции записи, обработки и распознавания звукового сигнала, группируются динамически подключаемые модули – прослойки для взаимодействия с другими приложениями и элементами операционной системы. Каждый такой модуль предоставляет основному приложению словарь команд, которые он способен обрабатывать. Кроме того, модуль уведомляет основное приложение о том, с чем именно он может взаимодействовать. Основное приложение отслеживает изменение фокуса в системе и в соответствии с этим формирует рабочий словарь как объединение словарей команд модулей, подписавшихся на работу с активным в данный момент приложением или элементом управления. Схематически перемещение потоков данных в такой системе соответствует изображенному на рис. 1. Жирные линии обозначают перемещение данных, тонкие – управляющих команд.

Преимущества такого подхода очевидны: сторонние разработчики могут создавать модули взаимодействия с интересующими их приложениями, не изменяя кода основного приложения-распознавателя; пользователь может по своему усмотрению

формировать набор активных модулей; создатели коммерческих проектов, желающие оснастить свой программный продукт функциями голосового управления, избавлены от необходимости открывать спецификации.

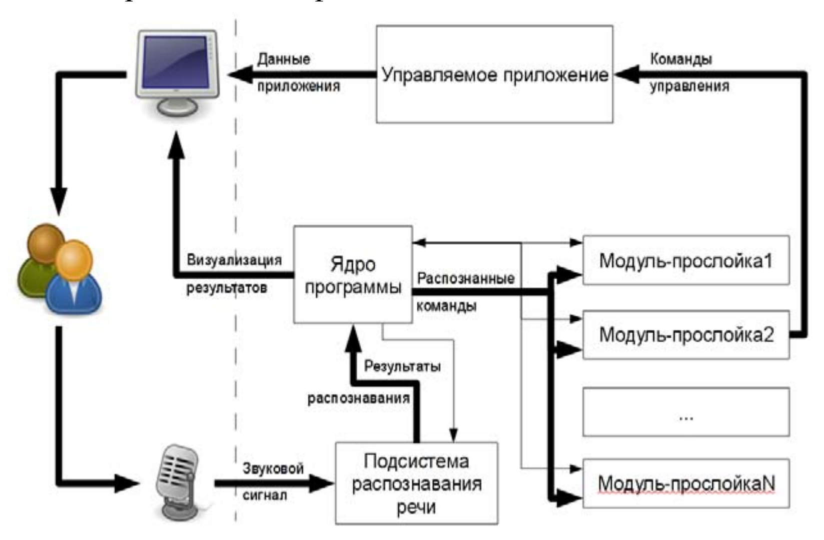


Рисунок 1 – Схема потоков данных системы голосового управления компьютером

Модуль-прослойка для голосового управления неким приложением представляет собой динамически присоединяемую библиотеку (*.dll), экспортирующую функции, описание которых приведено в табл. 1.

Словари, используемые системой распознавания, представляют собой текстовые файлы, каждая строка которых содержит информацию об одном распознаваемом классе. Такая строка должна содержать следующие поля, разделённые пробелами:

– символьное обозначение команды;
– слово, которое необходимо произнести для активации команды;
– обобщённая транскрипция;
– обозначение части речи («с» – существительное, «п» – прилагательное, «г» – глагол и т.д.).

Входящие в состав системы модули позволяют использовать голосовые команды при работе с редакторами формул «Equation» и «MathType», а также при работе с редактором формул из пакета Microsoft Office 2007. Возможности голосового управления в большинстве случаев освобождают пользователя от обращения к системе меню редактора формул, тем самым облегчая процесс ввода формул, делая его менее монотонным и трудоемким. При организации системы голосовых команд мы исходили из предположения, что символы, которые не требуют использования сочетания клавиш, пользователю проще вводить с клавиатуры, нежели голосом (все латинские символы, а также все цифры и символы типа *, &, +, -, = и так далее).

Таблица 1 – Функции, экспортируемые модулем-прослойкой

№ п/п	Название	Тип результата	Список аргументов	Описание
1.	GetAppName	void	char *cCom	Возвращает содержание заголовка окна, с которым может работать данный модуль. Вызывается основной программой каждый раз при изменении фокуса ввода в системе.

Продолж. табл. 1

2.	GetDictName	void	char *cCom	Возвращает путь к файлу, содержащему словарь команд. Вызывается в случае, когда система определила, что данный модуль может работать с получившим фокус ввода приложением.
3.	GetCommand	void	int nCom, char *cCom	Получить символьное обозначение команды с номером nCom.
4.	GetHint	void	int nCom, char *cHint	Получить подсказку (расширенное описание) для команды с номером nCom.
5.	GetWord	void	int nCom, char *cCom	Получить слово, которое должен произнести пользователь, чтобы была выполнена команда с номером nCom.
6.	GetCount	int	–	Получить количество команд, поддерживаемых модулем.
7.	CommandSequence	void	char *lpCom	Инициировать реакцию модуля на распознанное слово lpCom.
8.	DelCommand	void	char *lpCom	Отменить последнее действие, предпринятое как результат распознавания слова lpCom.

Организация подсистемы распознавания речи

Программа-распознаватель использует пофонемное распознавание на уровне широкой фонетической классификации (гласная – голосовая согласная – шипящая – пауза) [1], [2]. В основе лежит ряд разработанных авторами алгоритмов [3]. Для записи звукового сигнала производится его 8-битная оцифровка с частотой дискретизации 22 050 Гц. Программа использует автоматическое определение начала и конца речи и осуществляет проверку записанного на речь путем определения наличия или отсутствия достаточно длинных квазипериодических частей, которые должны отвечать голосовым звукам.

Предполагается использование системы в лабораторных условиях, при отсутствии существенного внешнего шума. Перед началом работы производится настройка автоматической записи. С микрофона записывается 30 000 отсчетов «тишины», и в записанном сигнале анализируются последовательные отрезки по 300 отсчетов в каждом. Для каждого из них вычисляется отношение:

$$V / C, \tag{1}$$

где

$$V = \sum_{i=0}^{298} \left| x_{i+1} - x_i \right| \tag{2}$$

– численный аналог полной вариации, C – число точек постоянства, то есть дискретных моментов времени, для которых в следующий момент величина сигнала остается неизменной. Автоматически определяется значение величины (1), характерное для используемой звуковой карты, как наиболее часто встречающееся в массиве значений. Оно увеличивается на 0,1 и используется в качестве порога срабатывания автозаписи, а результат, увеличенный в 2 раза – в качестве порога завершения записи.

В системе распознавания речи по нажатии кнопки записи компьютер начинает записывать сигнал, поступающий с микрофона, и вычислять для последовательных отрезков по 300 отсчетов величину (1), определяется момент, после которого эта величина впервые не менее пяти раз подряд превышает порог срабатывания и, начиная с него, в буфер 1 заносятся отсчеты вплоть до момента, после которого на протяжении 10 000 отсчетов величина (1) окажется меньше, чем порог завершения. После этого запись останавливается.

Описанная процедура записи звука может начаться под действием постороннего шума. Поэтому необходима проверка записанного на наличие речи. Практически любое слово содержит голосовые звуки, которым в записанном сигнале отвечают отрезки квазипериодичности. Исключение, например, – отдельно произнесенный предлог «С». На этом основан следующий способ проверки записанного на наличие речи. Задается некоторый порог p. Для мужского голоса его можно взять равным 100, для женского – 70. Если в записанном сигнале обнаружится не менее 5 идущих подряд квазипериодов с длиной p, то записанный сигнал содержит речь. В таком случае он передается в буфер 2 для последующей визуализации и распознавания. В противном случае этот сигнал во внимание не принимается.

После записи сигнала выполняется его автоматическая сегментация на участки, отвечающие классам широкой фонетической классификации.

Пусть имеется одномерный числовой массив и задан некоторый порог p. Построим символьную последовательность S, поставив в соответствие членам массива, которые больше p, символ «В» (выше порога), остальным – символ «Н» (ниже порога). Для того, чтобы устранить случайные единичные включения, для каждого промежуточного i-го элемента полученной символьной последовательности S выполняются две дополнительные обработки, обработка «тройками»:

если $s[i-1]=s[i+1]$ и $s[i]\neq s[i-1]$,

то полагается

$$s[i]=s[i-1],$$

и обработка «четверками»:

если $s[i]=s[i+3]$ и $s[i+1]\neq s[i]$, $s[i+2]\neq s[i]$,

то полагается

$$s[i+1]=s[i] \text{ и } s[i+2]=s[i].$$

Далее мы не раз будем иметь дело с описанной процедурой построения такой символьной последовательности. Для краткости будем называть ее «В-Н»-обработкой исходного числового массива.

Назовем сглаживанием сигнала

$$y_1, y_2, \dots$$

обработку его 3-точечным скользящим фильтром

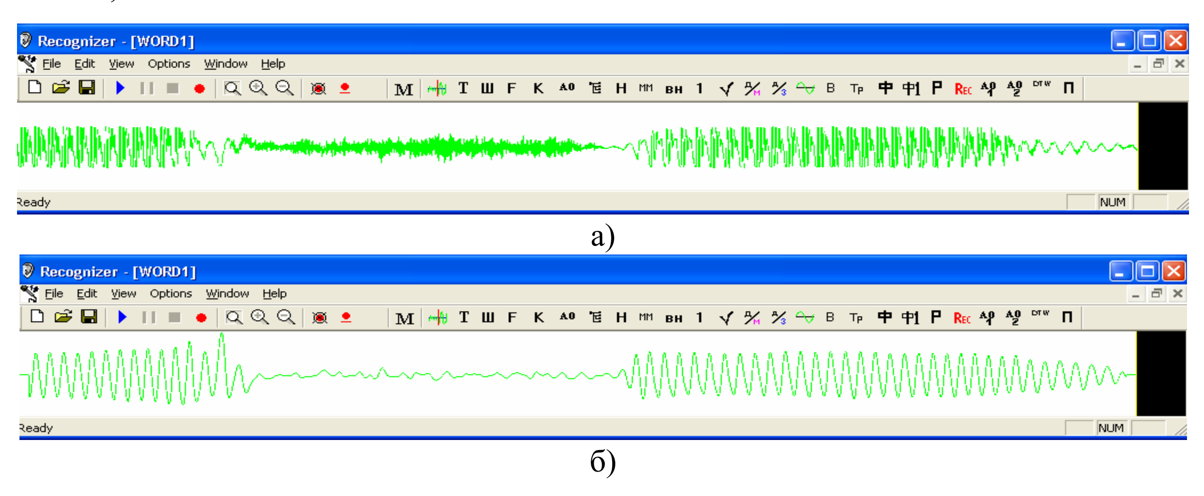
$$y_i=\frac{y_{i-1}+y_i+y_{i+1}}{3}.$$

Далее предлагается алгоритм выделения глухих согласных, произнесение которых происходит без участия голосовых связок. В основе его лежит обработка сигнала полосовым фильтром с интервалом пропускания от 100 до 200 Гц. На рис. 2 а) и 2 б)

приведена запись слова «ОСА» до и после такой фильтрации. Отфильтрованный сигнал пронормирован так, чтобы его максимальное значение равнялось 255 (либо минимальное значение равнялось нулю).

После фильтрации на участках глухих звуков разность между числом точек непостоянства и числом точек постоянства будет отрицательной, что позволяет выделить их в массиве таких разностей, построенном для последовательности окон в 256 отсчетов. На рис. 3 изображено окно программы, реализующей описанный алгоритм для слова «оса». Слева приведены столбцы трех числовых массивов: массив чисел точек постоянства, массив чисел точек непостоянства и массив разностей.

Подчеркнем еще раз, что любой выделенный участок глухих звуков может содержать шипящую, как в слове «лошадь», паузу, как в слове «лапа», или и то, и другое вместе, как в слове «отстать».



а)

б)

Рисунок 2 – Визуализация слова «оса» до и после фильтрации

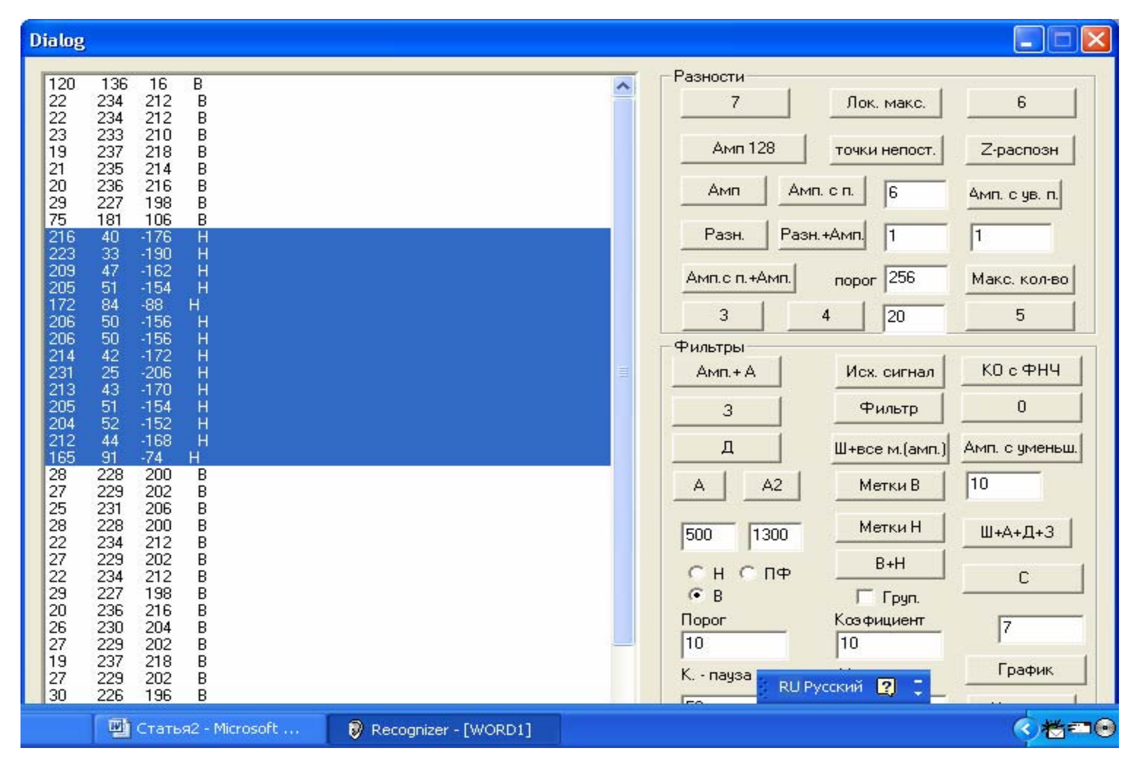


Рисунок 3 – Числовой массив, по которому определяются *С* в слове «оса»

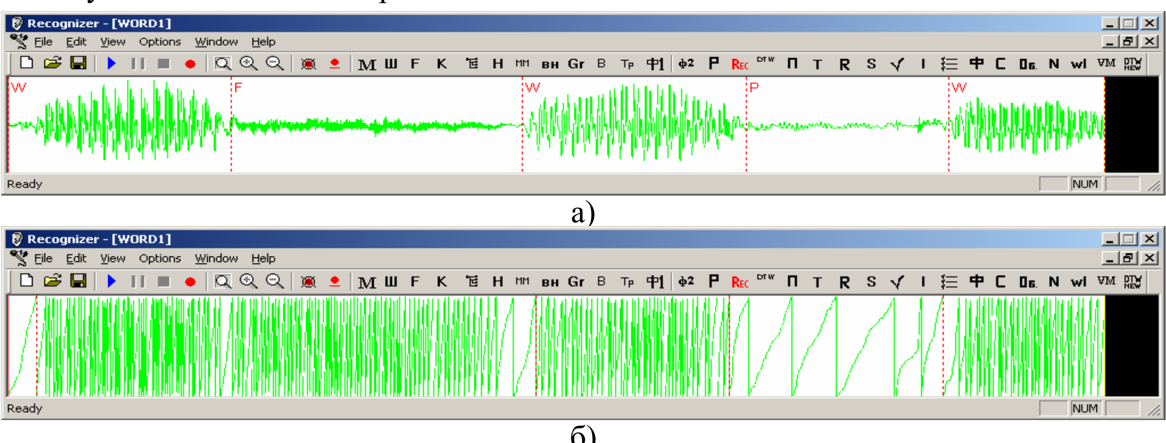
Рассмотрим для произвольно выделенного участка речевого сигнала численный аналог полной вариации «с переменным верхним пределом»:

$$V(0)=0, \quad V(n)=\sum_{i=0}^{n-1}\left|x_{i+1}-x_i\right|. \tag{3}$$

Пусть N_1 – максимальное число такое, что $V(N_1)\le 255$. Полагаем $W(n)=V(n)$ при $0\le n\le N_1$, $W(N_1+1)=0$, $W(n)=\sum_{i=N_1+1}^{n-1}\left|x_{i+1}-x_i\right|$ при $N_1+1\le n\le N_2$, где N_2 – максимальное число, такое, что $W(N_2)\le 255$ и так далее. В результате возникает массив чисел

$$N_1, N_2-N_1, N_3-N_2 \ldots \tag{4}$$

Каждое из них – это длина участка, на котором величина $W(n)$ возрастает от 0 до 255. Возьмем среднее этих чисел для выделенной части сигнала. Это среднее условимся называть «вариационной мерой» или просто «мерой» M выделенной части сигнала. На рис. 4 а) и 4 б) изображен график звукового сигнала и функции $W(n)$, построенной по нему описанным выше образом.



а)

б)

Рисунок 4 – Сигнал, отвечающий слову «осока» и график функции $W(n)$

На сегменте шипящей величина (3) быстро растет, поэтому числа (4) относительно малы. На сегменте паузы величина (3) растет медленно и поэтому числа (4) относительно велики. Для различения шипящей и паузы введем порог p (в системе авторов он взят равным 200). Возьмем выделенный сегмент глухих согласных и построим для него последовательность чисел (4). Те участки, для которых числа (4) превосходят p, относим к паузе (их объединение маркируем символом p), остальные – к шипящей (маркируем ее символом F). В результате компьютер расставит маркированные границы шипящих и пауз, как на рис. 4 а).

Теперь рассмотрим случай слова, состоящего только из голосовых звуков и не содержащего *Ж* и *З*. Разобьем сигнал на окна по 256 отсчетов и на каждом из них вычислим значение вариации

$$V=\sum_{i=0}^{254}\left|x_{i+1}-x_i\right|. \tag{5}$$

Далее от начала слова берется интервал из 20 таких окон и вычисляется среднее значение соответствующих величин (5), которое принимается за порог. Производится «В-Н»-обработка числового массива с этим порогом. Затем интервал, на котором выполняется описанная процедура, сдвигается вправо на одно окно и так далее. В результате возникает таблица вида, изображенного на рис. 5.

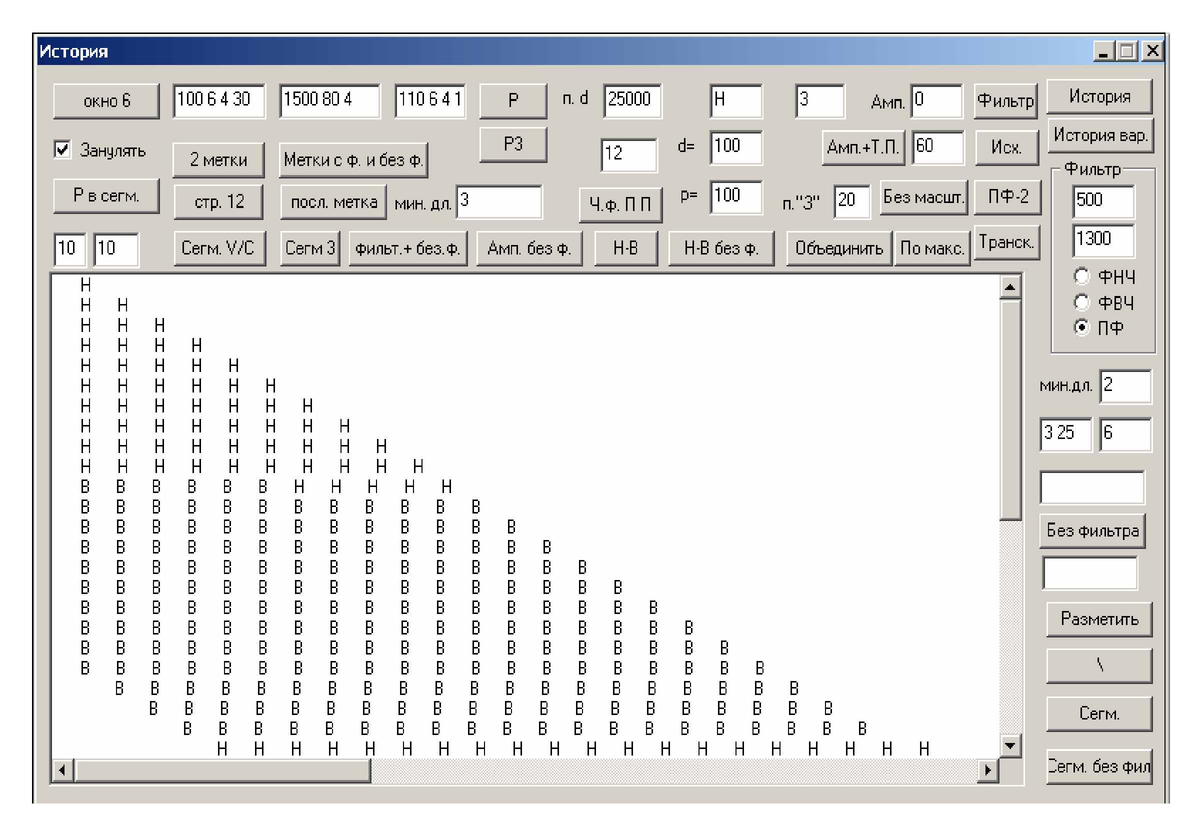


Рисунок 5 – Таблица, используемая при сегментации

Затем просматриваются все строки полученной таблицы и создается новая символьная последовательность S. Если текущая *i*-я строка таблицы начинается и заканчивается одним и тем же символом («Н» или «В»), то в S на *i*-ю позицию записывается соответствующий символ. Иначе считается количество вхождений каждого из символов в данной строке. Если количество «В» превышает количество «Н» или равно ему, то в S на соответствующую позицию записывается «В», иначе «Н». К полученной последовательности применяется «В-Н»-обработка. Метки сегментации ставятся там, где происходит смена символов «Н» на «В», или «В» на «Н». В-участок считается соответствующим гласной (возле левой метки проставляется символ W). Н-участок считается соответствующим голосовой согласной (возле левой метки проставляется символ C).

Если слово содержит шипящие или паузы, то мы выделяем их, как описано выше, после чего значения величины (3) для соответствующих им окон полагаем равными нулю и сегментируем сигнал только что описанным способом (шипящие и паузы автоматически попадают в число Н-участков). Для надежного выделения голосового согласного порядок формирования S непосредственно после шипящего или паузы меняется: если в строке появляется «В», но она заканчивается на «Н», то ей сопоставляется «Н». Дальше все по-старому. Аналогичная ситуация с голосовым согласным непосредственно перед шипящей или паузой.

Определенная трудность возникает при выделении в ходе сегментации участков согласных *Ж*, *З*. Они содержат существенную шумную компоненту (произносятся с той же артикуляцией, что и *Ш*, *С* и отличаются от них добавлением голоса). Поэтому для них значения величины (3) являются относительно большими (существенно превосходящими ее значения для чисто голосовых согласных) и они могут не попасть при обычной сегментации в число «Н»-участков. Участки, соответствующие указанным фонемам, целесообразно определять заранее, используя модифицированный алгоритм сегментации.

Для того чтобы перевести искомые участки в число согласных, для которых величина (3) мала, сигнал подвергается 5-кратному сглаживанию. После этого он сегментируется описанным выше способом. Для участков *Ж* и *З* характерна относительно большая вариация и значительное ее уменьшение при 5-кратном сглаживании. Поэтому выделенные с применением сглаживания «Н»-участки целесообразно промаркировать, вычисляя для каждого полученного «Н»-участка среднее значение величин (3) для поточечной разности исходного и пятикратно сглаженного сигнала. Если оно превышает некоторый порог p, участок считается маркированным, в противном случае – нет. Для одного из авторов и его оборудования эффективным оказывается порог $p = 200$.

Описанное сглаживание сигнала перед сегментацией помогает таким образом выделить участки *Ж* и *З* как «Н»-участки, но приводит к значительному ухудшению результата при разделении *М* и *И*. Поэтому целесообразно проводить окончательную сегментацию для исходного сигнала, сохранив полученную информацию относительно границ маркированных участков. Именно после того, как маркированные участки выделены, величины (3), отвечающие соответствующим окнам, полагаются равными нулю. В результате маркированные участки оказываются при работе с таблицей, изображенной на рис. 5, автоматически включенными в число «Н»-участков и можно провести сегментацию исходного сигнала в соответствии с описанным выше общим алгоритмом. Если при этом возникают метки, отстоящие от уже существующих меток для маркированных участков на расстояние не более чем в три окна в 256 отсчетов, они рассматриваются как лишние и удаляются.

Мы не касаемся в это статье выделения звука «*Р*». Отметим лишь, что при описанной процедуре сегментации он чаще всего выделяется как голосовой согласный.

Обучение системы для конкретного диктора

Реализованные в системе методики как пофонемного, так и пословного распознавания предполагают настройку системы на диктора. Обучение пофонемного распознавателя особенностям произношения пользователя заключается в создании дополнительных обобщённых транскрипций для каждой команды по результатам сегментации. Обучение для пословного распознавания подразумевает создание эталонов произносимых слов. Первоначально результатом распознавания обычно является список кандидатов, который в алфавитном порядке выводится в нижней части окна программы (рис. 6).

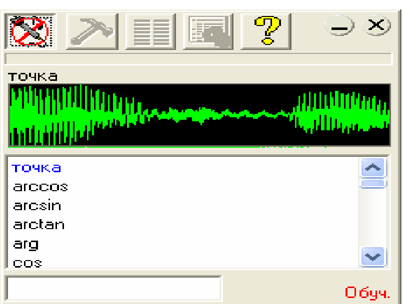


Рисунок 6 – Результат распознавания команды «точка»

Если одно или несколько слов соответствуют распознанной обобщённой транскрипции, то они выводятся перед остальным словарём, также отсортированные в алфавитном порядке и окрашенные в синий цвет. Среди этих слов происходит пословное распознавание, и наиболее близкое к эталону слово принимается в качестве окончательного решения и выносится на первое место в списке результатов.

Первоначально система является полностью необученной. Это означает, что каждому слову в словаре поставлена в соответствие только одна, «идеальная», обобщённая

транскрипция, соответствующая фонетическим правилам русского языка и отсутствуют эталоны слов-команд. После произнесения очередной команды пользователь может либо согласиться с результатом распознавания, либо откорректировать его, выбрав нужный вариант в списке кандидатов. При этом программа создает голосовой эталон сказанного слова, который в дальнейшем позволяет автоматически выделять нужный вариант из вышеупомянутого списка кандидатов. Ошибочно созданные эталоны удаляются автоматически после выполнения пользователем коррекции результатов. Таким образом, система позволяет сразу набирать формульный текст и в процессе работы постепенно обучается «понимать» пользователя без его дополнительного вмешательства.

Выводы

В статье рассмотрена архитектура модульной системы голосового управления, рассмотрены требования к функциям, экспортируемым модулем-прослойкой, а также требования к формату файла словаря команд. Такая модульная структура позволяет сосредоточить все функции голосового управления внутри одной программы. Кроме того, такой подход позволяет создавать модули-прослойки для управления конкретными приложениями авторам этих приложений или сторонним разработчикам.

Описанные методики определения границ речи в сигнале, выделения в сигнале, содержащем речь участков, соответствующих глухим звукам, шипящим, паузам, а также методики окончательной сегментации с одновременным распознаванием по широким фонетическим классам позволяют довольно точно получать обобщённые транскрипции распознаваемых слов, что позволяет значительно сократить словарь для дальнейшего распознавания слова как единого целого.

Предложенная в статье методика обучения системы в процессе использования позволяет сделать процесс настройки незаметным для пользователя и свести его к обычным операциям над системой голосового управления.

Литература

1. Шелепов В.Ю. Новые алгоритмы сегментации речевого сигнала и распознавания некоторых классов фонем / В.Ю. Шелепов, А.В. Ниценко // Искусственный интеллект. – 2007. – № 1. – С. 213-224.
2. Шелепов В.Ю. Новые алгоритмы распознавания фонем и их классов, поиск слова по его смешанной транскрипции при распознавании слов большого словаря / В.Ю. Шелепов, А.В. Ниценко, А.В. Жук // Искусственный интеллект. – 2007. – № 2. – С. 139-147.
3. Шелепов В.Ю. О распознавании фонем с помощью анализа речевого сигнала в частотной и временной областях. Приложение к распознаванию синтаксически связных фраз / В.Ю. Шелепов, А.В. Ниценко, А.В. Жук, Д.С. Азаренко // Речевые технологии. – 2008.– № 2. – С. 43-52.

В.Ю. Шелєпов, А.В. Ніценко, О.В. Жук

Побудова системи голосового керування комп’ютером на прикладі завдання голосового набору математичних формул

Статтю присвячено опису модульної системи голосового керування комп’ютером. Розглянуто питання, пов’язані з архітектурою такої системи, структурою керуючого модуля, форматом словника команд, наведено опис алгоритмів, що лежать в основі підсистеми розпізнавання мовлення. Розглянуто підхід до калібрування системи, який передбачає навчання в процесі використання системи.

V. Shelepov, A. Nicenko, A. Zhuk

Computer Voice Control System Development on Example of Mathematical Formula Voice Input Task

The article is devoted to computer modular voice control system description. The questions are examined, which are connected with architecture of such a system, controlling module structure and command dictionary format. The algorithms are described, which had formed the base of speech recognition subsystem. Also the approach to calibration of described system is offered, which supposes implicit calibration during system usage.

Статья поступила в редакцию 02.07.2010.